UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/638,491	08/15/2000	Michael Feldman	N298.12-0001	1274

164 7590 10/11/2005

KINNEY & LANGE, P.A.
THE KINNEY & LANGE BUILDING
312 SOUTH THIRD STREET
MINNEAPOLIS, MN 55415-1002

EXAMINER	
INGBERG, TODD D	

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 10/11/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/638,491	FELDMAN, MICHAEL	
	Examiner	Art Unit	
	Todd Ingberg	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 August 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-53* is/are pending in the application.

4a) Of the above claim(s) *1-26* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *28-55* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *8/15/2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claims 1 – 25 have been canceled.

Claims 26 – 27 have been withdrawn.

Claims 28 – 55 have been examined.

Claims 28, 43 and 49 have been amended.

Specification

The Examiner requests that claims 26 – 27 be canceled from this application. Currently. the claims are withdrawn.

Drawings

The drawings are objected because the shading of the following figures is too dark for patent literature and the label for Figure 2 is hand written. Shading is an issue for Figures 2, 3, 4, 5, 6, New formal drawings are required.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 28 – 53 are rejected under 35 U.S.C. 102(b) as being anticipated by Template Software Inc.

The **Template** product line contains:

The SNAP programming language (One manual used)

The Workflow Template (Two manuals used)

The Web Component (Not used in this Office Action)

These three layered products work together.

The documentation sets for the products contains the following manuals.

SNAP released June 1997

SNAP Language Reference (Not used in this Office Action)

Using the SNAP Language (Not used in this Office Action)

Using the SNAP Communication Component (Referred to as **COM**)

Using the SNAP Graphic User Interface Component (Not used in this Office Action)

Getting Started with SNAP (Not used in this Office Action)

Using the SNAP Display Editors (Not used in this Office Action)

SNAP Class Library Reference (Not used in this Office Action)

Using the SNAP External Application Software Component (Not used in this Office Action)

Using the SNAP Development Environment (Not used in this Office Action)

SNAP Module Library Reference (Not used in this Office Action)

Using the SNAP Permanent Storage Component (Not used in this Office Action)

Workflow released September 1997

Developing a WFT Workflow System (Referred to as **WFT**)

Using the WFT Development Environment (Referred to as **Using**)

WFT Library Reference (Not used in this Office Action)

Web Component

Using the Web Component (Not used in this Office Action)

Training Manuals

Workflow Template Training Course Version 8.0, 1997 (Section B, Referred to as **Train**).

Since, these products work together they constitute a single reference and can be used as the basis for a rejection based on anticipated by a product offering. Furthermore, with the 1997 press release announcing version 8.0 these considered prior art under ***In re Epstein*** 31 USPQ2d 1817 (decided August 17, 1994) with a 1997 release date despite the 1998 copyright date.

Claim 28

Template anticipates an architecture for developing a distributed information system (Template , Workflow System as described in the manuals made of record, **Using** , Chapter 3, Overview), the architecture comprising: a component development tool for generating a plurality of **autonomous** components (**Using,** Workflow Design Editor, Chapter 3 and Task Editor, Chapter 6), that implement and consume services (**Using**, Object Oriented Implementation – messaging is inherent in object technology, page 4-41, functions and **WFT**, messaging from objects to object both local on the same machine or on the net – Note one of ordinary skill in the art knows messages are the result of using methods – called functions in Template manual, page 7-23); ; a system development tool for defining a plurality of component instances based on the plurality of components (**Using** , instantiation of objects from the classes linked by inheritance, page 4-20), configuring the plurality of component instances, and defining links between component instances, without requiring writing of code (Ability to instantiate another instance of an object does not require additional code just execution of existing code- Instantiation is inherent in object technology) ; and an engine software (WFT, every where an object can be instantiated is a software engine) program to provide a dynamic run-time environment for hosting the plurality of component instances and supporting communication between component instances (**COM** , communications support, page 4-8) **based upon the defined links (Using, pages 2-42 to 43, and pages 3-5 to 3-15, and WFT page 2-16)** .

Examiner's Response

A good overview of the Workflow system is found in WFT Chapter 2). Both the WFT and Using manuals document the commercial product by Template System.

Claim 29

The architecture of claim 28, further comprising: a service definition tool for generating service protocols that are implemented by components, the service protocols defining a format of messages to be sent between ports of component instances (**Using**, Object Oriented Implementation – messaging is inherent in object technology, page 4-41, functions and **WFT**, messaging from objects to object both local on the same machine or on the net – Note one of ordinary skill in the art knows messages are the result of using methods – called functions in Template manual, page 7-23); and wherein links defined between component instances are

defined between the ports of component instances (WFT , flows 3-25 to 3-34 between nodes, pages 4-13 to 4-19).
Examiner's Response
The methods that send messages and receive messages and the ability to define the communications among objects.

Claim 30
The architecture of claim 28, wherein the component development tool provides the capability of representing components as a first and a second plurality of components, each component in the first plurality of components representing a physical entity in the distributed information system, and each component in the second plurality of components representing a logical entity in the distributed information system. (**Using,** Applications, Chapter 7 and 7-3, figure 7-1).
Examiner's Response
WFT also page 2-9 shows the logical view of the workflow system. Note the ROLES of Employee, Manger etc. WFT, page 7-56 to 7-62 covers distributed workflow systems, Chapter 7 covers the deployment which covers the physical topology of the network of the Workflow system.

Claim 31
The architecture of claim 29, wherein each of the ports comprises either a service provider port or a service consumer port (**Using** , tasks to perform tasks on work flow item are contained on consumer node, see Chapter 6 for Task Editor and chapter 7 for the Application Editor and COM, getters and setter, objects in object technology page 4-20 , 4-29 to 4-31).

Claim 32
The architecture of claim 31, wherein service provider ports and service consumer ports based on the same service protocol are complementary.
Examiner's Response
Messaging is inherent in object oriented technology and should have been well known to one of less than ordinary skill in the art well before the time of invention. When Applicant states the "same service protocol are complementary", the Examiner regards the word "complementary" as the word inherent. One can not take messaging out of object technology it is part of object oriented technology by definition.

Claim 33
The architecture of claim 32, wherein the system development tool only allows links to be defined between service provider ports and complementary service consumer ports (COM, getters and setter, objects in object technology page 4-20 , 4-29 to 4-31).
Examiner's Response
A getter message invokes a different method than a setter message.

Claim 34
The architecture of claim 28, wherein each of the plurality of component instances is self-sufficient.

Examiner's Response

An object by definition. The lifecycle of an object being instantiated until it is destroyed. Applicant has made claim to an inherent aspect of object technology.

Claim 35

The architecture of claim 34, wherein the only dependencies between component instances are logical dependencies (**Using**, flows between Applications, as per claim 4, pages 3-25 to 3-34 and **WFT** , flows 3-25 to 3-34 between nodes, pages 4-13 to 4-19).

Examiner's Response

The logic is how the nodes communicate with each other. Messages are sent among objects.

Claim 36

The architecture of claim 33, wherein a component instance includes at least one service provider port that allows multiple simultaneous links with complementary service consumer ports (**WFT** , ability for a node to communicate with multiple nodes, page 7-23 Peer to Peer).

Claim 37

The architecture of claim 33, wherein component instances are executed concurrently (**Using ,** executing the methods of an object concurrently, page 4-41), and wherein the communications between service provider ports and complementary service consumer ports are asynchronous (**COM,** where communications over net is asynchronous, pages 4-13 and page 11-4 and Appendix B – connection oriented) .

Examiner's Response

Applicant has made claim to another inherent aspect of object technology.

Claim 38

The architecture of claim 28, wherein the engine software program runs on each of a plurality of networked nodes (**Using,** deployment editor, Chapter 8 and **WFT**, page 7-1 to 7-7).

Examiner's Response

The Template system is a distributed Workflow system. The Application editor is designed to define the abilities of nodes.

Claim 39

The architecture of claim 38, wherein the system development tool represents the distributed information system as a single entity, regardless of physical node and network composition into which the component instances will be deployed (**WFT**, page 2-4, Figure 2-1, the actual deployment of the different ROLES is viewed as a single entity even thou the ROLES can be distributed on different nodes)

Examiner's Response

The Workflow Design Editor (WDE) is intended to be used by the domain expert (**Train** Section B, page 5).

Claim 40

The architecture of claim 38, wherein the system development tool deploys each component instance to one of the plurality of networked nodes. (**Using** , deployment editor, Chapter 8 – see page 8-10).

Claim 41
The architecture of claim 38, and further comprising a local repository on each of the plurality of nodes, the local repository on each node storing data defining the component instances deployed to and hosted by that node and storing link data for the component instances deployed to and hosted by that node (**Using** , deployment editor, Chapter 8 – see pages 8-10 to 8-12 commands in editor).

Claim 42
The architecture of claim 39, wherein the system development tool allows changes to be made to the component instances deployed to and hosted by the plurality of networked nodes and allows changes to be made to links between the component instances deployed to and hosted by the plurality of networked nodes. As per claim 41.

Claim 43
The architecture of claim 42, wherein the system development tool allows changes to be made to the component instances and allows changes to be made to links between the component instances, without requiring writing of additional code, wherein the system development tool allows the changes to be made while the distributed information system is running.
Examiner's Response
As per claims 28 and 42.

Claim 44
The architecture of claim 43, wherein the system development tool allows deletion of the component instances deployed to and hosted by the plurality of networked nodes and allows deletion of communication links between the component instances deployed to and hosted by the plurality of networked nodes, wherein the system development tool allows the deletions to occur while the distributed information system is running. As per claim 41, deployment editor.

Claim 45
The architecture of claim 29, and further comprising a central system repository for storing the components, the component instances, link data, infrastructure configuration and configuration data for the service protocols.
Examiner's Response
As per claim 28 and WFT, Chapter 7, Deploying Server.
Claim 46
The architecture of claim 28, wherein at least one of the component instances supports continuous activities internally. An object as per claim 34.

Claim 47

The architecture of claim 28, wherein each of the component instances is configurable to participate in activities that are collectively performed by multiple component instances.
Examiner's Response

As per claims 28 and 41. The defining of a Workflow System and deploying it.

Claim 48

The architecture of claim 28, wherein the only dependencies between component instances that are linked to each other are logical dependencies implemented using the component development tool. As per claim 41, deployment editor.

Claim 49

Template anticipates an architecture for developing a distributed information system(As per claim 28), the architecture comprising: a component development tool for generating **autonomous** components that implement and consume services(As per claim 28); a system development tool for defining a plurality of component instances based on the plurality of components(As per claim 28), configuring the plurality of component instances, and defining links between component instances(As per claim 28); and an engine software program to provide a programmable run-time environment for hosting the plurality of component instances(As per claim 28) and implementing the links to provide bi-directional communication paths between the plurality of component instances(**WFT** , ability for bi-directional communication paths between linked ports is a result of peer to peer communications, page 7-23 Peer to Peer).

Claim 50

The architecture of claim 49, further comprising a service definition tool for generating service protocols which define a format of messages to be sent through a plurality of ports, each port being associated with a component instance.
Examiner's Response

Messages and getters and setters as per claims 28 and 31.

Claim 51

The architecture of claim 49, wherein the engine software program provides the bidirectional communication paths between linked ports (**WFT** , ability for bi-directional communication paths between linked ports is a result of peer to peer communications, page 7-23 Peer to Peer).

Claim 52

The architecture of claim 49, further comprising a plurality of networked nodes running the engine software program, wherein the engine software dynamically manages ports and links (WFT , flows 3-25 to 3-34 between nodes, pages 4-13 to 4-19) for the component instances across the plurality of networked nodes. (**COM**, page 4-8 ability to handle messaging and object transmission, **WFT**, messaging from objects to object both local on the same machine or on the net – Note one of ordinary skill in the art knows messages are the result of using methods – called functions in Template manual, page 7-23).

Claim 53
The architecture of claim 49, wherein the component development tool is designed to be operated by a person skilled in computer programming.
Examiner's Response
Train, page 5 states the domain expert does not need to know about SNAP. SNAP is a programming language. The SNAP programmer does need to be skilled in computer programming.

Claim 54
The architecture of claim 49, wherein the system development tool is designed to be operated by a person without skill in the art of computer programming.
Examiner's Response
As per claim 53

Claim 55
The architecture of claim 49, wherein any component instance having a consumer port that complies with a first service protocol may be configured to communicate with any component instance having a provider port that also complies with the first service protocol.
Examiner's Response
Messages as per claims 28 and 29.

Response to Arguments

Applicant's arguments filed August 3, 2005 have been fully considered but they are not persuasive. The following are the Applicant's remarks as scanned.
"REMARKS
This is in response to the Final Office Action dated April 7, 2005 in which claims 28-55 were rejected under 35 U.S.C. § 102 as being anticipated by Template Software. With this Amendment, claims 28, 43 and 49 are amended. Claims 28-55 are pending in this application.

Claim Correction
It is respectfully requested that the amendment to claim 43 be entered to correct the spelling of the word "architecture."

Claim Rejections Under 35 U.S.C. § 102
Claim 28 is amended to make a number of clarifications. First, the claim amendment clarifies that the components generated by the component development tool are "autonomous." Second, the claim clarifies that the specified features of the system development tool do not require writing of any code. Third, the claim clarifies that the engine software program enables communication between component instances based upon the defined links.
With this Amendment, claim 28 is allowable over the prior art of record. The first element of amended claim 28 recites a component development tool for generating a plurality of autonomous components that implement and consume services. Autonomous components of the present invention are described, for example, at page 8, lines 9-15 (with reference, to FIG. 3),

which explains that the components of the present invention can "interact while being self-sufficient," and that components are "designed to operate as stand-alone entities, with no knowledge of peers or proxies, executing their own activities."

Template software does not disclose a tool for generating autonomous components. Rather, Template Software's approach is to generate code to define all aspects of the desired system and then compiling the code into a monolithic application. Evidence of the compiled nature of the system is found on page 8-20 of the document "Using the WFT Development Environment," where it says "You must build a business process node or server before you can run it." `Build" is defined on page 8-3 of the same document as "To compile the source files... into object files that can be executed. Building a business process node or server is the same as building an application that runs in the node or server." As a result, it can be seen that Template Software does not utilize autonomous components, because all relationships in the system are compiled. Rather, the complete system model must be defined up-front before the build. Furthermore, any subsequent changes to the system require a complete re-build and redeployment of the system. Therefore, Template Software does not disclose autonomous components as recited by claim 28. As a result, claim 28 is allowable over the prior art of record.

The second element of amended independent claim 28 recites a system development tool for defining a plurality of component instances based on the plurality of components, configuring the plurality of component instances, and defining links between component instances, without requiring writing of code.

The amendment clarifies that the system development tool does not require writing of code. Although a developer must write code and compile the code to create a component, code need not be written in the system development tool to utilize the component as defined in the claim.

In discussing the second element of independent claim 28, the Office Action stated that defining component instances, configuring component instances, and defining links between component instances, all without requiring writing of additional code were "basic concepts of object-oriented technology." This is not correct-the features of the present invention are not merely basic concepts of object-oriented technology. First, the Office Action states that defining component instances is the same as defining an object. However, the act of defining an object requires writing at least one line of code to define the object. Second, the Office Action states that configuring component instances is the same as writing methods that get and set values. However, writing methods is also writing code. Third, the Office Action states that defining links between component instances is one of the inherent aspects of object technology, namely messaging. However, messaging between objects also requires writing of code. Template Software's messaging between objects is described in more detail below. Since each of these features is done through the writing of code in the Template Software system, Template Software does not teach or suggest a system development tool for defining component instances, configuring component instances, and defining links between component instances without requiring writing of code.

The third element of amended independent claim 28 recites an engine software program to provide a dynamic run-time environment for hosting the plurality of component instances and supporting communication between component instances based on the defined links. The amendment clarifies that communication between component instances is based on the defined links. This feature of the present invention is related to the previously described feature of having

autonomous components. Since the components are autonomous, they are stand-alone entities, with no knowledge of peers of proxies. (P. 8, lines 9-15.) When a component is created the developer defines the messages that the component is capable of receiving on its incoming ports. The developer then defines the logic as to how to respond to those incoming messages, and defines the messages that the component will send out of its outgoing ports.
The component is compiled without defining the source of incoming messages or the recipients of outgoing messages. Links, which are uncompiled relationships between components can be established or modified at anytime while the component is running, without components being aware of the existence of those links.
A link that is defined between two components means that a message coming from the output port of one component will be delivered to the incoming port of another component, without either component having any knowledge of the other. The message is simply delivered, without the source expecting a response.
Components can exist without any links to other components, although in this case they will have no incoming messages. Un-compiled relationships between autonomous components allows a high level of flexibility as these relationships can be established, modified or deleted at anytime. Additional components can be introduced into the system at any time, while the system is running. These new components may be linked to any other compatible component in the system, providing extensibility/modification to the system functionality on the fly.

Template Software does not disclose an engine software program to provide a dynamic run-time environment for hosting the plurality of component instances and supporting communication between component instances based on the defined links. Rather, as described above, Template Software uses a method of generating code to define all aspects of the desired system and then compiling the code into a monolithic application. Since Template Software does not teach each and every element of claim 28, claim 28 is in condition for allowance. Reconsideration and notice to that effect is respectfully requested.

With this Amendment, independent claim 49 also is amended to clarify that components are autonomous. This feature of the present invention, as described above, is not disclosed by Template Software. Based upon this and other features defined in the claim and explained above that are not disclosed by Template Software, independent claim 49 is in condition for allowance. Reconsideration and notice to that effect is respectfully requested."

Examiner's Response

Applicant states the Template Software commercial product fails to support the following:

A. "Autonomous components "

B. "..specified features of the system development tool do not require writing of any code. "

C. "... the engine software program enables communication between component instances based upon the defined links."

The Template system does support autonomous components the routing between the Applications (Nodes or Roles) is separate from the actual build and deployed object oriented system. This means the Applicant's description of *uncompiling* the routing information is not accurate. The Applications (Nodes or Roles) are built and deployed. They are implemented using object-oriented technology. The links called routing and routing rules are stored in separate CD

files and are designed in the Workflow Development Environment (WDE) an interpreted environment. The hard coded links the Applicant states exist do not in fact exist. The rejection above for the first independent claim points to the section of the reference where a list of separate C files are listed. The modular and flexible implementation is not "a monolithic application" as the applicant has characterized. This would be a mischaracterization of the actual for use and for sale product of Template Software Inc.

Applicant's Remarks

Dependent claims 29-48 and 50-55 depend from allowable independent claims 28 and 49 respectfully and are therefore allowable.

Examiner's Response

For the same reasons the independent claims are not allowable the dependent are held in rejection.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Todd Ingberg whose telephone number is (571) 272-3723. The examiner can normally be reached on during the work week..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Todd Ingberg
Primary Examiner
Art Unit 2193

TI